第六届胖头鱼杯信息学蒟蒻争霸赛

时间：2026 年 4 月 11 日 14：00 至 17：30，共 3.5 小时。

题目名称	鸡巧	鸡腿	鸡算	鸡友	鸡划
题目类型	传统型	传统型	传统型	传统型	传统型
目录名称	skill	leg	calculate	friends	plan
源程序文件名	skill.cpp	leg.cpp	calculate.cpp	friends.cpp	plan.cpp
输入文件名	skill.in	leg.in	calculate.in	friends.in	plan.in
输出文件名	skill.out	leg.out	calculate.out	friends.out	plan.out
单测试点时限	1 秒	1 秒	1 秒	1 秒	1 秒
内存限制	256 MB	256 MB	256 MB	256 MB	256 MB

注意事项与提醒（请选手务必仔细阅读）：

1. 参赛选手需要用 `玩家ID` 作为名称创建文件夹，按照 `源程序文件名` 创建每道题的程序文件，并**直接**放在 `玩家ID` 文件夹中。可以参考试题目录下的 `PTYB-J000` 文件夹，也可以直接将该文件夹重命名成自己的 `玩家ID`。
2. 评测时以 `输入文件名` 和 `输出文件名` 作为名称进行文件输入输出。可以参考试题目录下的 `PTYB-J000` 文件夹中的样例程序。
3. 所有名称必须使用**英文小写**。
4. 结果的比较方式为全文比较（忽略**行末空格**和**文末换行**）。
5. 程序可使用的栈内存空间限制与题目的内存限制一致。
6. 每道题都提供了 `Linux` 系统格式的样例文件，放在试题目录下名为 `sample` 的文件夹中。
   注意，使用 `Windows` 的 `记事本` 打开样例文件时**可能**会出现格式错误，可以用 `写字板` 或者编程软件打开。
7. 评测环境为 `NOI Linux2.0`，评测机配置为 `Intel(R) Core(TM) i7-8700K CPU @3.70GHz`，编译选项为 `-O2 -std=c++14 -static`，允许使用万能头文件 `<bits/stdc++.h>`。

鸡巧（skill）

题目描述

本套题目一共有 5 道题，按顺序编号为 $0 \ldots 4$，即题目【鸡腿】编号为 1 、【鸡算】编号为 2 、【鸡友】编号为 3 、【鸡划】编号为 4 。

每道题的测试数据都被分成了几个部分，在【数据说明】段落有详细的阐述。有些数据拥有**特殊属性**，可以采用针对性的算法求解并获得这些分数，所以解题时不一定需要解决完整的问题，用较少的时间获得部分的分数也是不错的选择。

每道题的总分均为 100，请仔细阅读所有题目并回答，如果正确计算了编号为 n 的题目中第 m 部分的测试数据可以获得多少分?

输入格式（文件名：skill.in）

只有一行，包含两个整数 n, m，分别表示题目的编号和测试数据部分的序号。

输出格式（文件名：skill.out）

只有一行，输出可以获得的分数。

样例数据

【样例 1 输入】

```
2 3
```

【样例 1 输出】

```
10
```

【样例 1 解释】

查看题目【鸡算】的【数据说明】部分可知，正确计算该部分的测试数据可以获得 10 分。

> - 对于额外 10% 的测试点，满足 $a_i = a_0$ 。样例 3 符合当前性质。

数据说明

对于所有测试点，满足 $1 \leq n \leq 4$ 、 $1 \leq m \leq t_i$，其中 t_i 表示编号为 i 的题目的测试数据的分档数量。

鸡腿（leg）

题目描述

在大海的深处，居住着一只英俊的胖头鱼，他非常向往胖头鸡的生活，所以向坤神许下了心愿，以两年半不准打篮球的代价，换取了一双胖头鸡的腿。

他打算去往胖头鸡的部落拜访朋友，以胖头鱼出发的位置作为原点 $(0,0)$ 建立平面直角坐标系，则胖头鸡的部落在 (n,m) 。假设某时刻胖头鱼所在的位置为 (x,y) ，为了避免迷路，他每次只会**向右**移动**一个**单位到达 $(x+1,y)$ 或者**向上**移动**一个**单位到达 $(x,y+1)$ 。可惜胖头鱼还不能很熟练的使用鸡腿，所以他在不同位置进行移动时需要消耗的体力是不同的。具体来说，当 $x+y$ 是**偶数**时，向右移动一个单位需要消耗 a 的体力，向上移动一个单位需要消耗 b 的体力；当 $x+y$ 是**奇数**时，向右移动一个单位需要消耗 c 的体力，向上移动一个单位需要消耗 d 的体力。

胖头鱼想知道，为了到达胖头鸡的部落，他**最少**需要消耗多少体力。由于他实在没有力气思考了，所以只好求助于你。

输入格式（文件名：leg.in）

只有一行，包含 6 个整数，依次为 n,m,a,b,c,d ，表示坐标以及在不同位置进行移动时需要消耗的体力。

输出格式（文件名：leg.out）

输出一行，只有一个整数，表示最少需要消耗的体力。

样例数据

存放在 `sample/leg` 目录下。

【样例 1 输入】

存放在 `leg1.in` 文件中。

```
11 0 1 2 3 4
```

【样例 1 输出】

存放在 `leg1.out` 文件中。

```
21
```

【样例 1 解释】

只能不断的向右移动，其中 6 次在 $x+y$ 为偶数的位置， 5 次在 $x+y$ 为奇数的位置，消耗的体力为 $6\times 1+5\times 3=21$ 。

【样例 2 输入】

存放在 `leg2.in` 文件中。

```
5 3 2 1 5 3
```

【样例 2 输出】

存放在 `leg2.out` 文件中。

```
22
```

【样例 2 解释】

最优的移动方式为 $(0,0) \to (1,0) \to (1,1) \to (2,1) \to (3,1) \to (4,1) \to (4,2) \to (5,2) \to (5,3)$ 。

消耗的体力为 $2+3+2+5+2+3+2+3=22$ 。

【样例 $3,4,5$ 输入与输出】

存放在 `leg3.in` 与 `leg3.out` 、 `leg4.in` 与 `leg4.out` 、 `leg5.in` 与 `leg5.out` 文件中。

数据说明

对于所有测试点，满足 $0 \le n,m,a,b,c,d \le 10^9$ 。

测试点的特殊属性如下：

- 对于前 20% 的测试点，满足 $a=c$ 且 $b=d$ 。
- 对于额外 20% 的测试点，满足 $a=b$ 且 $c=d$ 。
- 对于额外 20% 的测试点，满足 $a<b$ 且 $c>d$ 。样例 3 符合当前性质。
- 对于额外 20% 的测试点，满足 $a<b$ 、 $c<d$ 且 $n>m$ 。样例 4 符合当前性质。
- 对于额外 20% 的测试点，没有特殊属性。样例 5 符合当前性质。

鸡算（calculate）

题目描述

胖头鸡部落流行着 n 种货币，分别编号为 $0 \dots n-1$，货币之间的换算满足**混合进制**的规则，第 i 种货币的**进率**为 a_i，即 a_i 个编号为 i 的货币等价于 1 个编号为 $i+1$ 的货币。

使用这个混合进制，可以像十进制一样用一个**数字串**表示任意**非负整数**的数值，根据货币种类的编号**由大到小**写出每种货币的数量，并省略**前导** `0`。例如，有 2 种货币且 $a_0 = 7$，那么 `43` 对应的数值用十进制表示就是 31。

尴尬的是，胖头鱼根本不会混合进制，但是胖头鸡部落的所有计数都会使用这个进制！现在胖头鱼想要购买 c 个单价为 b 的礼物送给朋友，只好请你来帮忙计算一下总价格 s 了。

输入格式（文件名：calculate.in）

第一行包含一个正整数 n，表示货币种类的数量。

第二行包含 n 个正整数 $a_0, \dots, a_{n-1}$，按照编号**由小到大**给出每种货币的进率。

第三行包含一个数字串 b，表示用混合进制表示的礼物的单价。

第四行包含一个数字串 c，表示用混合进制表示的礼物的数量。

输出格式（文件名：calculate.out）

输出一个数字串 s，表示用混合进制表示的礼物的总价格。

样例数据

存放在 `sample/calculate` 目录下。

【样例 1 输入】

存放在 `calculate1.in` 文件中。

```
5
10 10 10 10 10
265
73
```

【样例 1 输出】

存放在 `calculate1.out` 文件中。

```
19345
```

【样例 1 解释】

所有数字串均采用十进制表示，直接用乘法计算可知 $265 \times 73 = 19345$。

【样例 2 输入】

存放在 `calculate2.in` 文件中。

```
5
7 5 9 4 6
536
45
```

【样例 2 输出】

存放在 `calculate2.out` 文件中。

```
51122
```

【样例 2 解释】

可以用十进制进行验证， `536` 对应的数值为 202， `45` 对应的数值为 33，相乘的结果为 $202 \times 33 = 6666$，表示成混合进制为 `51122` 。

【样例 $3, 4, 5$ 输入与输出】

存放在 `calculate3.in` 与 `calculate3.out` 、 `calculate4.in` 与 `calculate4.out` 、 `calculate5.in` 与 `calculate5.out` 文件中。

数据说明

对于所有测试点，满足 $1 \leq n \leq 10^5$ 、 $2 \leq a_i \leq 10$； b, s 不为 0 且长度不超过 n 、 c 的长度不超过 10；b, c, s 均满足进制数的规则，即每一位上的数字均小于该位的进率。

测试点的特殊属性如下：

- 对于前 20% 的测试点，满足 $n \leq 18$ 且 $a_i = 10$ 。
- 对于额外 20% 的测试点，满足 $a_i = 10$ 。
- 对于额外 10% 的测试点，满足 $a_i = a_0$ 。样例 3 符合当前性质。
- 对于额外 30% 的测试点，满足 $n \leq 18$ 。样例 4 符合当前性质。
- 对于额外 20% 的测试点，没有特殊属性。样例 5 符合当前性质。

鸡友（friends）

题目描述

胖头鱼打算拜访 n 个朋友，朋友的编号为 $1 \dots n$，为了方便，他每天会拜访一段**连续编号**的朋友。第 i 个朋友的热情度为 a_i，胖头鱼每天获得的热情度就是拜访的**所有**朋友的热情度的**总和**。胖头鱼计划用 k 天将每个朋友都**刚好**拜访一次，虽然他不会数论，但是他最近学会了一个很酷的词汇叫做**最大公约数**，所以他希望这 k 天获得的热情度的最大公约数尽可能大。

由于他还不确定需要花几天来拜访朋友，所以请对 1 到 n 中的每个整数 k，计算每天热情度的最大公约数的**最大值**。

输入格式（文件名：friends.in）

第一行包含一个整数 n，表示朋友的数量。

第二行包含 n 个整数 a_i，第 i 个整数表示第 i 个朋友的热情度。

输出格式（文件名：friends.out）

只有一行，输出 n 个整数，之间用一个空格隔开。第 i 个整数表示当胖头鱼计划用 i 天拜访完所有朋友时，每天热情度的最大公约数的最大值。

样例数据

存放在 `sample/friends` 目录下。

【样例 1 输入】

存放在 `friends1.in` 文件中。

```
6
1 1 1 1 1 1
```

【样例 1 输出】

存放在 `friends1.out` 文件中。

```
6 3 2 1 1 1
```

【样例 1 解释】

用 1 天时，热情度为 6；用 2 天时，可以每天拜访 3 个朋友，每天的热情度均为 3；用 3 天时，可以每天拜访 2 个朋友，每天的热情度均为 2；用不少于 4 天时，至少有 1 天的热情度为 1。

【样例 2 输入】

存放在 `friends2.in` 文件中。

```
6
1 2 3 4 5 6
```

【样例 2 输出】

存放在 `friends2.out` 文件中。

```
21 3 3 3 1 1
```

【样例 2 解释】

用 2 天时，可以每天分别拜访 $5, 1$ 个朋友，热情度为 $15, 6$；用 3 天时，可以每天分别拜访 $3, 2, 1$ 个朋友，热情度为 $6, 9, 6$；用 4 天时，可以每天分别拜访 $2, 1, 2, 1$ 个朋友，热情度为 $3, 3, 9, 6$。

【样例 $3, 4, 5$ 输入与输出】

存放在 `friends3.in` 与 `friends3.out`、`friends4.in` 与 `friends4.out`、`friends5.in` 与 `friends5.out` 文件中。

数据说明

对于所有测试点，满足 $1 \leq n \leq 10^5$、$1 \leq a_i \leq 10^7$。

测试点的特殊属性如下：

- 对于前 25% 的测试点，满足 $n \leq 20$。
- 对于额外 25% 的测试点，满足 $n \leq 5000$ 且 $a_i = 1$。样例 3 符合当前性质。
- 对于额外 30% 的测试点，满足 $n, a_i \leq 10000$。样例 4 符合当前性质。
- 对于额外 20% 的测试点，没有特殊属性。样例 5 符合当前性质。

鸡划（plan）

题目描述

胖头鱼在回到大海之前，还想邀请朋友一起去进行一次健身，胖头鱼有 n 个朋友，朋友的编号为 $1 \dots n$，为了方便，他打算邀请一段**编号连续**的朋友。

朋友在一起健身时，总是会聊一些爱好，一共有 m 种爱好，爱好的编号为 1 到 m，第 i 个朋友拥有 c_i 个爱好，分别为 $d_{i,1}, \dots, d_{i,c_i}$。

两个朋友如果**没有任何共同**的爱好，他们在健身时就不会进行交流。如果有至少 k 对朋友之间不会交流，那么这次健身的气氛就会变得有些尴尬。现在胖头鱼想知道，在保证健身氛围不尴尬的前提下，**最多**能够邀请多少个朋友，由于他只擅长健身，所以只好请你帮忙计算一下。

输入格式（文件名：plan.in）

第一行包含三个整数 n, m, k，表示朋友的数量、爱好的数量和使得气氛变得尴尬的最少的不交流的朋友对的数量。

接下来 n 行，每行首先包含一个整数 c_i，表示第 i 个朋友拥有的爱好数量；然后包含 c_i 个**互不相同**的整数 $d_{i,j}$，表示每个爱好的编号。

输出格式（文件名：plan.out）

输出一个整数，表示最多可以邀请的朋友数量。

样例数据

存放在 `sample/plan` 目录下。

【样例 1 输入】

存放在 `plan1.in` 文件中。

```
10 1 6
1 1
1 1
0
1 1
1 1
0
1 1
1 1
1 1
1 1
```

【样例 1 输出】

存放在 `plan1.out` 文件中。

```
6
```

【样例 1 解释】

可以邀请编号 4 到编号 9 的朋友，只有编号为 6 的朋友不会与其他人交流，其他朋友互相之间都会交流，所以共有 5 对朋友之间不会交流。

【样例 2 输入】

存放在 `plan2.in` 文件中。

```
10 2 2
0
1 1
2 1 2
1 2
0
2 1 2
1 2
1 1
0
2 1 2
```

【样例 2 输出】

存放在 `plan2.out` 文件中。

```
3
```

【样例 2 解释】

可以邀请编号 2 到编号 4 的朋友，只有编号 2 和编号 4 的朋友不会交流。

【样例 $3, 4, 5$ 输入与输出】

存放在 `plan3.in` 与 `plan3.out` 、 `plan4.in` 与 `plan4.out` 、 `plan5.in` 与 `plan5.out` 文件中。

数据说明

对于所有测试点，满足 $1 \leq n \leq 10^5$ 、 $1 \leq d_{i,j} \leq m \leq 10$ 、 $0 \leq c_i \leq m$ 、 $1 \leq k \leq n^2$ 。

测试点的特殊属性如下：

- 对于前 15% 的测试点，满足 $m = k = 1$ 。
- 对于额外 15% 的测试点，满足 $n \leq 100$ 。
- 对于额外 15% 的测试点，满足 $n \leq 1000$ 且 $m = 1$ 。样例 3 符合当前性质。
- 对于额外 10% 的测试点，满足 $n \leq 1000$ 且 $m = 2$ 。
- 对于额外 10% 的测试点，满足 $n \leq 1000$ 且 $m \leq 6$ 。样例 4 符合当前性质。
- 对于额外 15% 的测试点，满足 $m = 1$ 。
- 对于额外 20% 的测试点，没有特殊属性。样例 5 符合当前性质。